The Complexity of Finding an Optimal Policy for Language Convergence

Kiran Lakkaraju[1] and Les Gasser[2]

[1] Department of Computer Science
University of Illinois, Urbana-Champaign
klakkara@uiuc.edu
[2] Graduate School of Library and Information Science
University of Illinois, Urbana-Champaign
gasser@uiuc.edu

Abstract. An important problem for societies of natural and artificial animals is to converge upon a similar language in order to communicate. We call this the language convergence problem. In this paper we study the complexity of finding the optimal (in terms of time to convergence) algorithm for language convergence. We map the language convergence problem to instances of a Decentralized Partially Observable Markov Decision Process to show that the complexity can vary from P-complete to NEXP-complete based on the scenario being studied.

1 Introduction

Language is a collective property of the society. A language is inherently a communicative system (although it has some non-communicative interactions with agents, Clark ([1]) suggests that in addition to a communicative function, language can serve as a tool to reshape the computational space that our brains must handle), that allows agents to interchange information.

In this work we study how a set of initially diverse (in terms of languages) agents can come to an agreement upon a single language. We refer to this as the *language convergence* problem.

Previous work in this area has focused on the convergence rate of a particular algorithm. Each agent has a learning algorithm which will learn a language based on examples of sentences from other agents. The algorithm for convergence usually specifies a set of agents that each agent can interact with, and the parameters of the learning algorithm.

In this paper we want to explore the question, how hard is it for an agent to learn how to converge? We do not want to know how to converge in a specific setting, but rather how to converge in a whole set of situations. For instance, we want a *policy* for the agent that will tell it whom to interact with in order for the agent to be able to communicate with the entire society after a period of time.

Other work has focused on evaluating single algorithms to determine if, when an agent follows a specific policy, will the entire society converge. For instance,

S. Nolfi et al. (Eds.): SAB 2006, LNAI 4095, pp. 804–815, 2006.

Cucker, Smale and Zhou give bounds on how many other agents each agent must interact with in order for the entire population to converge, given a policy where each agent is to learn from the sentences it gathers of other agents languages ([2]). Steels creates a simulation and empirically shows the convergence of a population of agents in [3]

Our work differs from the above because we want to study the higher order problem of how hard it is to learn an algorithm for convergence, not how long it takes to converge using a particular algorithm. We want to find the optimal algorithm for convergence. The optimal algorithm, when implemented by an agent, will result in the quickest convergence. To study the complexity of finding an optimal algorithm for convergence we show how to map instances of language convergence problems to instances of a Decentralized Partially Observable Markov Decision Process (*Dec-POMDP*) [4]. The optimal algorithms for convergence correspond to the optimal joint policies of a *Dec-POMDP* . We make use of previous complexity results for finding the optimal joint policy for a *Dec-POMDP* ([4], [5])

By mapping language convergence scenarios to the *Dec-POMDP* model we can gain insight on the computational complexity of finding an optimal solution. This provides us with insight on the worst-case complexity of solving these language convergence scenarios.

In this paper, four language convergence scenarios are examined, single goal oriented, multiple goal oriented, teacher-student, and teacher-student with population observation. Each scenario can be modeled as a type of *Dec-POMDP* .

In Section 2 we go over the *Dec-POMDP* model and the mapping to the language convergence situation. Next, we examine four different language convergence scenarios mapped to the *Dec-POMDP* . Section 3 examines the complexity of the four different language convergence scenarios. Finally we talk about some related work, future work, and conclusions.

2 Language Convergence as a *Dec-POMDP*

In this section we describe the language convergence problem as a *Dec-POMDP* .

A *Dec-POMDP* is very similar to a POMDP except that in a *Dec-POMDP* the state changes based on the actions of multiple agents. In a *Dec-POMDP* , we have a set of agents embedded in an environment, modeled as a global state. The agents can execute actions that produce a change in the environment and possibly a reward. Each agent makes its own observation about the environment at each time step. In a POMDP there is only a single entity controlling the system. While the process is controlled by multiple agents, there is only one reward which is based on the single global state.

The structure of the behavior of the agents is:

1. Each agent, in parallel, observes the environment. This generates an observation for each agent.
2. Each agent, in parallel, chooses an action by using their policies and the observation they have just perceived.

3. The global state changes, based on the current global state, and the actions of every agent.
4. One reward is generated for all the agents based on the previous state, the actions executed, and the resulting state.

Formally, a *Dec-POMDP* is a tuple:

$$M =< S, A, P, R, \Omega, O, T >$$

where (assuming the number of agents is n):

– S is a finite set of states, with initial state s^0. The state at the current time will be called the *global state*.
– $A = \{A_i | A_i$ is a finite set of actions for agent $i.\}$
– P is a transition function, giving the probability $P(s'|s, a_1, \ldots, a_n)$ of moving from state s to state s', given actions $a_1, \ldots, a_n$, where a_i is the action executed by agent i.
– R is a global reward function, giving the system-wide reward $R(s, a_1, \ldots, a_n, s')$ when actions $a_1, \ldots, a_n$ cause the state-transition from s to s'.
– $\Omega = \{\Omega_i | \Omega_i$ is a finite set of observations for agent i $\}$
– O is an observation function, giving the probability $O(o_1, \ldots, o_n | s, a_1, \ldots, a_n, s')$ that each agent i observes o_i when actions $a_1, \ldots, a_n$ cause the state transition from s to s'. Where o_i is the observation of agent i.
– T is the time-horizon (finite or infinite) of the problem.

A joint policy $< \delta_1, \ldots, \delta_n >$, is a set of local policies, δ_i where

$$\delta_i : \Omega_i^* \rightarrow A_i \tag{1}$$

The joint policy specifies a policy for each agent that will determine the action an agent should take at each time step based on the sequence of observations it has made. Figure 1 is an illustration of the *Dec-POMDP* model.

See [5] for a full description of various classes of the *Dec-POMDP* . Roughly, we can characterize the various sub-classes of *Dec-POMDP* by how much of the global state each agent can observe (from each agent fully observing the global state, to each agent only observing its own "local" state) and the accuracy to which they can view the states (from viewing the state itself to viewing an observation of it). Different combinations of these properties induce different complexities when solving the *Dec-POMDP* .

A *Dec-POMDP* has independent transitions if the global state can be factored into n components such that the actions of an agent affects only its component. An independent transition *Dec-POMDP* will be referred to as an IT, *Dec-POMDP*

A *Dec-POMDP* has independent observations if the state can be factored into n components such that the observations of an agent depend only upon its component and the actions it has executed. An independent observation *Dec-POMDP* will be referred to as an IO,*Dec-POMDP*

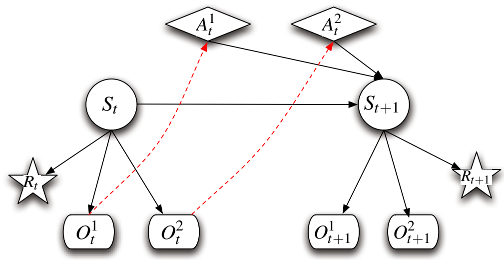

Fig. 1. Illustration of a *Dec-POMDP* . S_t represents the state of the environment at time t. A_t^1 and A_t^2 represent the actions that agents 1 and 2 executed at time t. R_t is the reward at time t, and O_t^1 and O_t^2 are the observations at time t. The dotted arrows between the observations and actions indicate that these observations were known when making the decision on which action to execute at the next time step.

Following [5], we assume that the same decomposition of the state holds for the independent transition and independent observations. We will refer to an agents component of the state as its *partial view* or its *local state*. The partial view of an agent will be denoted by S_i.

An example of an IO, IT *Dec-POMDP* is a simple gridworld situation. Suppose multiple agents are wandering around a 2-d gridworld. The state of the system would be the aggregate locations of each agent. The partial view of each agent would be its location. It is easy to see that any action an agent does (for instance "Move North") will only affect its own state, thus satisfying the independent transition property. We can further have each agent observe only its own location, thus satisfying the independent observation property.

If each agent can determine the global state based only on its sequence of observations, we say the *Dec-POMDP* is *Fully-Observable*. In our gridworld example, this would be like each agent knowing the locations of every other agent based only on its own observations.

If there is a mapping from the aggregate observations of every agent to the current state, then we say the *Dec-POMDP* is *Jointly Fully-observable* A jointly fully-observable *Dec-POMDP* is called a *Dec-MDP* . The gridworld example above is actually a *Dec-MDP* . The aggregation of each agents observations is, by definition, the state of the system.

If each agent can determine its local state from its sequence of observations, then we say the *Dec-POMDP* is *Locally Fully-observable.*

A finite horizon Goal-oriented *Dec-MDP* is a *Dec-MDP* with the following conditions (taken from [5]):

1. There exist a set of states $G \subset S$ of global goal states. At least one state of G must be reachable by some joint policy
2. The process ends at time T

3. All actions in A incur a cost, $C(a_i) < 0$. For simplicity we assume the cost is dependent only upon the action.
4. The global reward is $R(s, < a_1, \ldots, a_n >, s') = \sum_{i=1}^{n} C(a_i)$
5. If at time T the system is in a state $s \in G$ there is an additional reward $JR(s) \in \Re$ that is awarded to the system for reaching a global goal state.

A *GO-Dec-MDP* has *uniform cost* when the cost of all the actions are the same. There is also a NOP action that an agent can perform which has cost 0 and does not change the state.

2.1 Finding the Optimal Policy

The main question is, how hard is it to find a joint policy that maximizes the expected total return over the finite horizon? Bernstein et. al. in ([4]) have shown that deciding whether there exists a joint policy with at least a certain value, via an off-line algorithm, is NEXP-Complete for *Dec-POMDP* and *Dec-MDP* where $n \geq 2$.

The work in *Dec-POMDP* s has looked at finding a joint policy *offline*. This means that the model is known, and as many simulations as needed can be run. While during the search for the policy the model is known, during the execution of the policy the agents will not know the entire model. The joint policy that is to be found must take into account the constraints of the agents during execution of the policy.

Goldman et. al. study the complexity of various subclasses of the *Dec-POMDP* problem in [5]. Table 1 summarizes the results from [5]

Table 1. Complexity of *Dec-POMDP* and related models. The third column indicates where this result was obtained. The lemmas and section 3 refer to [5]. The NBCLG property will be examined in Section 2.3.

Model	Complexity	
Dec-POMDP	NEXP-C	[4]
Dec-MDP	NEXP-C	[4]
IO,IT Dec-MDP	NP-C	Lemma 4
IO, IT Dec-POMDP	NEXP	Section 3
GO-Dec-MDP	NEXP-C	Lemma 3
IO, IT, GO-Dec-MDP, 1 goal	P-C	Lemma 5
IO, IT, GO-Dec-MDP, NBCLG	P-C	Lemma 6

2.2 Mapping the Language Convergence Problem to a *Dec-POMDP*

The *Dec-POMDP* is an appropriate model to use to study the language convergence problem because the *Dec-POMDP* explicitly models decentralized control. There is a global goal - that of the entire population having the same language, but only local control - each agent independently decides what action to take. A *Dec-POMDP* explicitly models this situation, as there is a global reward based on the state of the system, yet the dynamics of the system are based on the aggregate decisions of each agent.

The first issue when mapping language convergence scenarios to *Dec-POMDP* is how to represent a language. We look at a simple type of language, represented as an association matrix. A language is considered as a mapping from meanings (objects, actions) to signals (words). There are many ways this mapping could be represented (for instance the mapping is a continuos function from the space of meanings to the space of objects in[2]) but one popular way is as an association matrix. The rows of an association matrix correspond to meanings, and the columns to words. An entry at row i and column j denotes the association between meaning i and word j. In this work we do not constrain languages to be represented as association matrices. We do, however, require that the number of different languages be finite. An association matrix can represent synonyms (multiple entries in a row) and homonyms (multiple entries in a column).

When representing a language as an association matrix, agents can change their language by modifying the association values in the matrix. In this work we assume that each agent has a finite set of actions that can modify its language.

In the language convergence case we want to reward the population when all the agents have the same language. We can do this by setting the state space to reflect the languages used by the agents. Let α be a set of n agents. Each agent can use a language from the finite set L of languages. At each point in time, every agent will be using a particular language from L. Let l_i denote the language of agent i. We set the state of the *Dec-POMDP* at time t to be the aggregate of the languages for each agent: $s_t =< l_1, l_2, \ldots, l_n >$. Thus the state set will be L^n.

An optimal joint-policy is a policy that, when the agents use it, will result in quick convergence to a high value state. We can formally specify this by setting the reward function to reward quick convergence. We constrain the reward function to be independent of the actions and the previous state, $R(s, a_1, \ldots, a_n, s') = R(s')$. Every state will have a small negative reward, except for the states where every agent has the same language. These states will have a high positive reward.

$$R(s' = \langle l_1^{t+1}, \ldots, l_n^{t+1} \rangle) = \begin{cases} 1 & if\ l_1^{t+1} = l_2^{t+1} = \ldots = l_n^{t+1} \\ -\epsilon & otherwise \end{cases}$$

Where l_i^{t+1} is the language used by agent i at time $t+1$ and ϵ is a small positive constant.

Under this reward function the policy that maximizes reward will minimize the number of states to get to a converged state from s_0.

We can also model situations in which specific languages have different rewards. As a means of communicating information, a language must be *effective* (allow agents to communicate all important meanings),*efficient* (computable, tractable) and *shared* (each agent must be able to understand each other).

We can assign to each language an objective value based on its effectiveness and efficiency. The reward function will give a higher reward to a state where all the agents have converged on a more effective and efficient language.

The set of actions that each agent can execute will depend on the type of languages that each agent can use. In this paper we will assume that each agent has the same action set, A. The transition probability function P will define the effects of executing an action.

The observations possible to an agent also have a large effect on the computational complexity of finding a solution. In this paper we will assume that each agent has the same set of observations, Ω.

2.3 Four Language Convergence Scenarios

In this section we go over four different language convergence scenarios. They differ from each other in terms of how much information each agent has about itself and the other agents in the population.

Single Goal Oriented. One of the simplest cases of a convergence problem occurs when there is only one language that has a positive reward. In this simple case, each agent can only observe its own language. In addition, let us assume that the action the agents can execute will change their language in some form, but the effect of the action does not depend upon the language of the other agents. In this situation each agent is moving in the state space trying to find the language that every agent will converge upon. This can be mapped to a uniform cost IO, IT *GO-Dec-MDP* .

An example of this situation would be where a language is represented as an association matrix where each row can only have a single entry. This means that each language does not contain any synonyms or homonyms. The set of actions would be the set of row swaps - that is we swap the meanings for two words.

Assume that s^g is the single global goal state that the agents want to converge to. Then since we are using an uniform cost *GO-Dec-MDP* , every state except s^g will have a negative reward.

Since the effect of the actions will only change the language of the agent that executed the action; and since the effect is determined only by the agents language and not on the other agents languages, the system satisfies the independent transition property.

Since the observations of a single agent depend only upon its own language the system satisfies the independent observation property.

Thus, we have a uniform cost, independent transition, independent observation, jointly-fully observable *GO-Dec-POMDP* . By Lemma 5 of [5] deciding this problem is P-Complete.

Intuitively this result makes sense, even though each agent only observes its own language. The policy for each agent can be determined independently of the policies for every other agent. Since it is known that all the agents will eventually reach the single goal state (since that is the only state that provides a positive reward), we can decompose this problem into n separate MDP's (assuming that the *Dec-POMDP* is locally fully observable).

Many Goal-Oriented. We can extend the Single Goal-Oriented case to involve multiple languages that the population can converge upon. This is more realistic since there are often multiple languages that a population can converge upon.

This situation is difficult because each agent might choose to pursue a different goal. Each agent will have to coordinate with the other agents to choose the same goal, which will lead to a very complex space of policies to search in.

Goldman and Zilberstein outline the *No Benefit to Change Local Goals* (NBCLG) property, which, if satisfied, will allow the system to be decomposed into a set of MDP. If a system satisfies the NBCLG property, it is P-Complete (Lemma 6 of [5]

The NBCLG property basically makes sure that it never is beneficial for an agent, while executing the optimal joint policy, to switch which goal state to go to. For instance, suppose that there is an optimal joint policy to one of the goal states. This can be computed for each agent by constructing a MDP for each agent to its component of the joint goal state. While executing the optimal joint policy an agent might veer from the optimal route (since the effects of actions are probabilistic, there is a chance that this might happen). If the NBCLG property is satisfied then even when an agent veers off the optimal route, it is guaranteed that the agent will not switch to another goal state.

Satisfying the NBCLG property depends upon the structure of the transition probability function. Verifying that a system satisfies this property would be quite difficult as well, since we would have to compute the value of changing goals at every intermediate step.

Teacher-Student. In many situations agents change their language via a *language game* ([6]). In a language game, a speaker and a hearer agent are drawn from the pool of agents. The agents interact with each other, exchanging words or sentences from their language. After the agents interact, either the hearer or both the speaker and hearer change their language based on the communicative success of the interaction.

What makes this situation different is that the language of an agent changes based on the language of another agent. In our previous examples, each agent modified its language independent of the languages of the other agents.

We can model this situation in a *Dec-POMDP* by having the actions correspond to the execution of a language game with a particular agent. There will be n actions, one for initiating a language game with each agent. The effects of these actions are to change the language based upon the language of the agent executing the action as well as the agent that is chosen to talk with.

In this case the independent transition property does not hold. The probability of an agents state at time $t + 1$ depends on both participants of the language game. The *Dec-MDP* still has independent observations though. Thus this situation can be modeled as an IO, *Dec-MDP* .

The complexity of an IO, *Dec-MDP* has not been studied yet. The complexity of an IO, *Dec-MDP* is bounded by the complexity of an IO,IT *Dec-MDP* (NP-Complete by Lemma 4 of [5]) and the complexity of a *Dec-MDP* (NEXP-Complete)

Teacher-Student with Population Observation. We can extend the previous Teacher-Student case by having each agent observe not just its own language, but the languages of the whole population. This situation will be mapped to a *Dec-MDP* .

Instead of having each agent observe its own language, we can allow the agents to sample the languages of the the entire population. In this case, the observations are dependent upon the state of the entire population and not just on the state of the current agent.

For instance, the observation of an agent might be of the language that is the most used. Or else, altering Ω to be the set of natural numbers the observation can be the number of agent using the same language as the observing agent. In either of these situations the observations depend upon the state of the population and not the partial view of the agent.

In this case, the language convergence problem is mapped to a *Dec-MDP* . The complexity of finding an optimal solution to a *Dec-MDP* is NEXP-Complete.

3 Complexity of Language Convergence

The four situations outlined above varied widely in terms of complexity. What makes the different situations easy or difficult to solve? The key is the level of uncertainty present in the system. There are two levels of uncertainty present, the first is the agents uncertainty of its own state, and the second is the agents uncertainty about the state of the other agents. Both of these factors affect the complexity of finding an optimal solution. Uncertainty about the agents state means that there will be an exponential number of possible policies that must be searched. Uncertainty about the state of other agents affects the size of the joint policy space that must be searched through.

In the general case the local policy of each agent will be a mapping from sequences of observations to actions. The policy must be from sequences of observations to actions because the agent is uncertain about the state that it is in. This means that there are $|A|^{\Omega^T}$ possible policies for the agent (where T is the finite horizon).

On the other hand, when the agent has knowledge of its state, the size of the policy can be substantially reduced. See [5] for more details.

While uncertainty about the local state of an agent affects the size of a policy, uncertainty about the state of other agents affects the number of policies that must be searched. If each agent knew the state of all the other agents then we could just model this as a MDP or POMDP and solve it. But since each agent does not know the state of the other agents we have to search through the combinations of policies.

In the single goal oriented case, each agent knew with certainty its current state. Lemma 1 of [5] proves that an IO,IT *Dec-MDP* is locally fully observable. This means that the size of the space of policies that need to be searched can be reduced because we don't have to consider all possible sequences of observations.

Rather, a policy for an agent will be a mapping from the local states of the agent to actions. This significantly decreases the size of the space of policies to search through.

In addition, there is no need to search through a joint policy space in the single goal oriented case. Since there is only one state with a positive reward, and each agent is striving to maximize reward, it is unnecessary to consider the policies of the other agents. It is guaranteed that at some point all the agents will reach the single goal state. Because of this assumption, finding an optimal joint policy reduces to finding n different policies, one for each agent. This is much less complex than searching for a single joint policy.

We can see that in the single goal oriented case there is no uncertainty about the local state of the agent and no uncertainty about the behavior of the other agent. Thus finding a solution is $P - Complete$.

The second situation, multiple goal oriented, is very close to the first situation except that we have added uncertainty about the state of the other agents. In the multiple goal case, the agents might converge upon different goal states, thus we cannot simplify the situation to finding n different policies.

If the *Dec-POMDP* satisfies the NBCLG property, though, it is like the single goal oriented case. Finding a policy for an NBCLG satisfying *GO-Dec-MDP* is similar to finding a policy for a single goal oriented *GO-Dec-MDP* . Since we know that once a goal is chosen no agent will veer from that goal, we are free to look at each goal state, and find the optimal policy for each agent to get to its partial view of the goal state. The goal state chosen will be the one which has the highest reward. Since we know the agents will never veer from going towards this goal state, we have found the optimal policy.

The third situation is another case where the agent does not know the state of the other agents, it is similar to the multiple goal oriented case.

The fourth situation, teacher-student with population observation, provides the most complex case. In this situation each agent does not know its own state, nor does it know the state of the other agents. Thus finding a policy is computationally expensive, since each policy will have to take into account all the possible sequences of observations, and all combinations of local policies will have to be considered.

4 Related Work

A good review of many Multi-Agent System models to the language convergence problem is given in [7]. There has been some work in studying the theoretical underpinnings of MAS models. Cucker, Smale, and Zhou [2] provide a mathematical formulation for a MAS simulation. In their work, each agent gets a set of example sentences from every other agent based on a pre-specified level of interaction between the agents. They investigate the number of examples each agent must be exposed to in order for the population of agents to converge.

[4] introduces and studies the complexity of finding optimal policies for the *Dec-MDP* and *Dec-POMDP* models. In this paper they show that deciding

these problems is NEXP-Complete. In other papers they present algorithms for constrained versions of these problems.

[5] studies various modifications of the *Dec-MDP* and *Dec-POMDP* models. These variations include goal directedness, communication, and independent transitions/observations. They showed the complexity of these problems as well as specified two algorithms for the goal directedness cases. In some cases, for instance when there is a single goal state and the transitions and observations of each agent are independent of each other, the problem becomes P-Complete.

5 Future Work

While the work here focuses on theoretical bounds for finding the optimal policy off-line, it would be very interesting to see if we can use some multi-agent reinforcement learning algorithms to learn an optimal policy.

This work shows that finding the optimal policy can be quite computationally expensive. This is because the specificity of the model is quite high - all actions every agent takes must be analyzed. On the other hand population based models like the Language Dynamical Equation ([8]) are much more tractable while giving up knowledge of the specifics of agents actions.

We are investigating approaches that incorporate the best of both worlds. The creation of a model that has the generalization and tractability of the LDE but also the fine-grained control and information that a MAS model can give us.

The crucial parameter in deciding the complexity of the language convergence problem is the amount of information that an agent has about the rest of the population. In the case of a fully observable *Dec-POMDP* , each agent can know the state of every other agent, and thus the problem can decompose into n independent MDP's. Direct communication is a possible way for agents to achieve full observability, but communication usually incurs a cost. This cost might be managed by specifying an interaction topology that limits the interaction between agents. Delgado, in ([9]), shows that a set of agents can agree on the same convention even when each agent might not interact with all the other agents. This work could provide a starting point for studying how limiting the interaction of agents could still result in language convergence.

A interesting avenue for future work would study how different interaction topologies for message passing affect the rate of convergence, and the complexity of finding optimal joint policies.

6 Conclusion

In this work we have investigated the complexity of finding an optimal policy for language convergence problems. Our main contribution is in mapping instances of language convergence problems to *Dec-POMDP* 's

Four examples of language convergence problems, and their associated *Dec-POMDP* 's were shown. In the simplest case, when there is only one language that all the agents can converge upon, deciding whether an optimal policy exists is P-Complete.

At the other extreme we have a situation where the agents are playing language games and each agent must decide who to interact with at every time step. In this case, when the agents cannot fully observe what language they are currently using, deciding upon an optimal policy is NEXP-Complete.

We have argued that the increase in complexity of finding an optimal policy is based on 2 levels of uncertainty, uncertainty over an agents local state and uncertainty over the state of the other agents in the population.

By mapping instances of the language convergence problem to instances of *Dec-POMDP* 's we have been able to study the worst case complexity of finding an optimal algorithm for the agents. This provides us with an intuition on what makes the language convergence problem complex. In future work we plan on adding communication between agents thus allowing them to gain knowledge of the languages used by other agents.

References

1. Clark, A.: Magic words: How language augments human computation. In Carruthers, P., Boucher, J., eds.: LANGUAGE AND THOUGHT: INTERDISCIPLINARY THEMES. Cambridge University Press: (1998)
2. Cucker, F., Smale, S., Zhou, D.X.: Modeling language evolution. Foundations of Computational Mathematics **4**(3) (2004) 315–343
3. Steels, L.: The origins of ontologies and communication conventions in multi-agent systems. Autonomous Agents and Multi-Agent Systems **1**(2) (1998) 169–194
4. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of markov decision processes. Math. Oper. Res. **27**(4) (2002) 819–840
5. Goldman, C., Zilberstein, S.: Decentralized control of cooperative systems: Categorization and complexity analysis. Journal of Artificial Intelligence Research **22** (2004) 143–174
6. Steels, L.: The evolution of communication systems by adaptive agents. In Alonso, E., D.K., Kazakov, D., eds.: Adaptive Agents and Multi-Agent Systems: Adaptation and Multi-Agent Learning. LNAI 2636, Springer Verlag, Berlin (2003) 125–140
7. Wagner, K., Reggia, J., Uriagereka, J., Wilkinson, G.: Progress in the simulation of emergent communication and language. Adaptive Behavior **11** (2003) 37–69
8. Komarova, N.L.: Replicator-mutator equation, universality property and population dynamics of learning. Journal of Theoretical Biology **230**(2) (2004) 227–239
9. Delgado, J.: Emergence of social conventions in complex networks. Artificial Intelligence **141**(1-2) (2002) 171–185